The Journal of Open Source Software

chessboard: An R package for creating network connections based on chess moves

Nicolas Casajus [1¶], **Érica Rievrs Borges** [1], **Éric Tabacchi** [2], **Guillaume Fried** [3], and **Nicolas Mouquet** [1,4]

1 FRB-CESAB, Montpellier, France **2** CNRS, Toulouse, France **3** ANSES, Montpellier, France **4** MARBEC, Univ Montpellier, CNRS, IFREMER, IRD, Montpellier, France ¶ Corresponding author

DOI: 10.21105/joss.05753

Software

- Review
- Repository
- Archive

Editor: Nikoleta Glynatsi

Reviewers:

- @Pentaonia
- @gsapijaszko

Submitted: 13 July 2023
Published: 13 October 2023

Summary

chessboard aims to facilitate the creation of connectivity matrices for sampling networks designed as regular grids. It can handle directed (asymmetric) and undirected (symmetric) spatial (or non-spatial) network connections. chessboard offers various methods to detect neighbors, all based on the chess game, allowing the creation of complex connectivity scenarios.

Statement of need

The analysis of network connections is present in many research fields, whether the network represents spatial connections or not, and is based on different proposed methods for defining neighbors. In ecology, spatial network analyses are widely used to investigate spatial patterns of organisms' distribution and provide important information about the underlying ecological processes leading to these patterns (Legendre & Legendre (2012), Pilosof et al. (2017)). Such methods rely on building a connectivity matrix (also known as an adjacency matrix) among sampling units (hereafter 'nodes'), which consists of a square matrix of dimension $n \times n$ (where n is the total number of nodes). This connectivity matrix thus represents the presence or absence of a link (hereafter 'edge') between each pair of nodes. When building a connectivity matrix, for each node, the detection of neighbors (new edges) will depend on the type of sampling design used (surface area, transect, regular grid, irregular grid, Dray et al. (2006)).

Here we introduce chessboard, an open-source R package (R Core Team (2023)), dedicated to facilitating the process of creating connectivity matrices for samplings designed as regular grids (i.e., the distance between each pair of nodes along one axis is always the same). chessboard identifies neighbors based only on node position on a two-dimension non-spatial referential. It can handle spatial networks, but it does not explicitly use geographical coordinates to find neighbors (it is not based on spatial distance). chessboard can handle directed (asymmetric) and undirected (symmetric) spatial (or non-spatial) networks.

Different methods are available in chessboard to detect neighbors, all based on the chess game as proposed in many R packages like spdep (Bivand & Wong (2018)), but with the advantage of several additional chess movements that are flexible with respect to the user's experience and that allow the creation of complex connectivity scenarios (Figure 1). chessboard implements the following rules to detect neighbors and to create edges:

- the **degree** of neighborhood: the number of adjacent nodes that will be used to create edges;
- the **orientation** of neighborhood: can neighbors be detected horizontally, vertically, and/or diagonally?

- the **direction** of neighborhood: does the sampling have a main direction? This can be particularly relevant for directed networks (e.g., rivers).

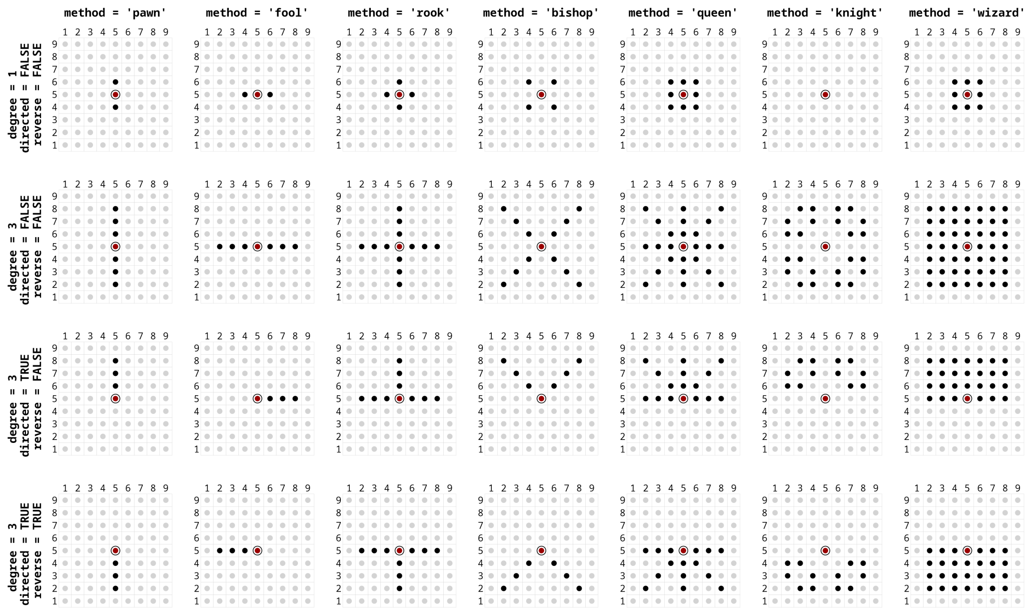

Figure 1: Overview of methods available in `chessboard` to detect neighbors. Red dots locate the node of interest (5-5) and black dots correspond to detected neighbors. Each column corresponds to a specific method derived from the chess game. Each row illustrates the use of one argument (row 1: default settings; row 2: use of the argument `degree`; row 3: use of the argument `directed`; row 4: use of the argument `reverse`)

`chessboard` provides simple visualization functions and generates outputs that are compatible and reusable with various other R packages: `adespatial` (Dray et al. (2023)), `spdep` (Bivand & Wong (2018)), `igraph` (Csardi & Nepusz (2006)), `sf` (Pebesma (2018)), and `ggplot2` (Wickham (2016)).

Main features

This section is an overview of the main features of `chessboard` (Figure 2).

The Journal of Open Source Software

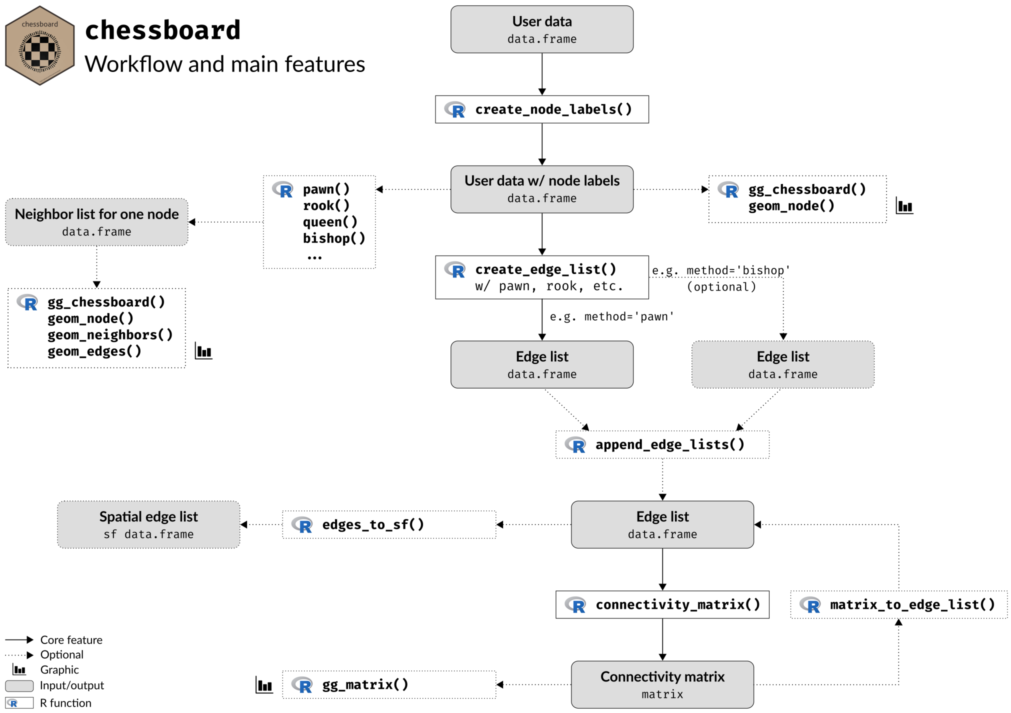

Figure 2: Main features and usage of the R package `chessboard`.

To illustrate the package, let's create a fictitious sampling design of dimensions 5 x 5 (25 nodes in total). By convention, we will name the x-axis ***transects*** and the y-axis ***quadrats***.

```
# Fictitious sampling (non-spatial) ----
sampling <- expand.grid("transect" = 1:5,
                        "quadrat"  = 1:5)
```

When working with `chessboard`, the first step is to create labels for nodes with the function `create_node_labels()`.

```
# Create node labels ----
nodes <- create_node_labels(data     = sampling,
                            transect = "transect",
                            quadrat  = "quadrat")
```

Node labels are a combination of the transect identifier (i.e., the position of the node on the x-axis of the grid) and the quadrat identifier (i.e., the position of the node on the y-axis of the grid).

Then we implement a connectivity scenario where we would like to connect nodes according to the 'bishop' move , i.e., diagonally, with a degree of neighborhood of 2 for an undirected network. The function `bishop()` (and all other functions named after the chess game) returns a subset of the nodes object (`data.frame`) containing the neighbors of the focus node.

```
# Find neighbors according to the bishop move (for one node) ----
nb <- bishop(nodes    = nodes,
             focus    = "2-3",
             degree   = 2,
             directed = FALSE)
```

Plotting functions available in `chessboard` can be used to inspect the results (Figure 3):

```r
gg_chessboard(nodes) +
  geom_edges(nodes, "2-3", nb) +
  geom_neighbors(nodes, nb) +
  geom_node(nodes, "2-3")
```

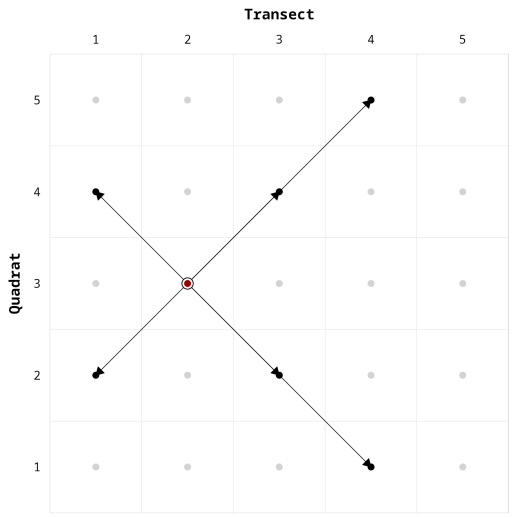

Figure 3: Neighbors and edges detected for one node using the bishop method and a degree of neighborhood of 2 (undirected network). The red dot locates the node of interest (2-3) on the chessboard. Black dots correspond to detected neighbors and arrows are corresponding edges.

The Chess pieces vignette details all possible moves implemented in `chessboard` and the effects of the arguments `degree`, `directed`, `reverse`, and `self`.

The function `create_edge_list()` will detect the neighbors for the 25 nodes:

```r
# Create edges according to the bishop move (for all nodes) ----
edges <- create_edge_list(nodes    = nodes,
                          method   = "bishop",
                          degree   = 2,
                          directed = FALSE)
```

This function returns an **edge list**, i.e., a two-column `data.frame` where a row corresponds to an edge between two nodes.

The function `connectivity_matrix()` computes the connectivity matrix of this undirected network.

```r
# Build connectivity matrix ----
mat <- connectivity_matrix(edges)
```

Finally, the function `gg_matrix()` can be used to plot the resulting matrix (Figure 4).

```r
# Plot connectivity matrix ----
gg_matrix(mat)
```

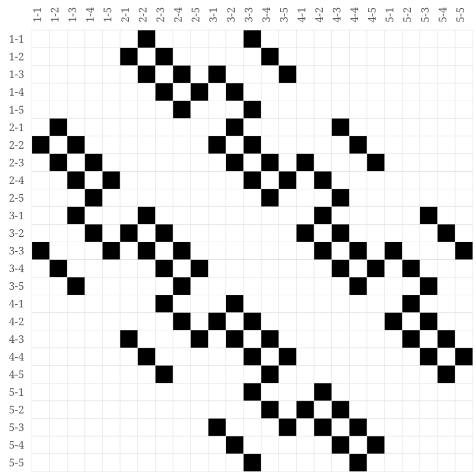

Figure 4: Connectivity matrix of a 5 transects x 5 quadrats network when edges (black squares) are created using the bishop method and a degree of neighborhood of 2 (undirected network). Each row and each column of this square matrix corresponds to a node of the network.

`chessboard` provides three long-form documentations to learn more about the package:

- a Get started vignette describing the core features of the package;
- a Chess pieces vignette detailing the different methods available to detect neighbors;
- a Visualization tools vignette describing the plotting functions available in `chessboard`.

Acknowledgements

This work is a product of the BRIDGE research project funded by the French national research program ITTECOP (CILB, MTE, ADEME and OFB) and the French Foundation for Research on Biodiversity (FRB) through its Centre for Synthesis and Analysis of Biodiversity data (CESAB).

References

Bivand, R., & Wong, D. (2018). Comparing implementations of global and local indicators of spatial association. *TEST*, *27*, 716–748. https://doi.org/10.1007/s11749-018-0599-x

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*, *1695*, 1–9. https://igraph.org/

Dray, S., Bauman, D., Blanchet, G., Borcard, D., Clappe, S., Guénard, G., Jombart, T., Larocque, G., Legendre, P., Madi, N., & Wagner, H. H. (2023). *adespatial: Multivariate multiscale spatial analysis*. https://cran-r-project.org/package=adespatial

Dray, S., Legendre, P., & Peres-Neto, P. R. (2006). Spatial modelling: A comprehensive framework for principal coordinate analysis of neighbour matrices (PCNM). *Ecological Modelling*, *196*, 493. https://doi.org/10.1016/j.ecolmodel.2006.02.015

Legendre, P., & Legendre, L. (2012). *Numerical Ecology (3rd Edition)* (p. 1006). Elsevier Science BV. https://www.elsevier.com/books/numerical-ecology/legendre/978-0-444-53868-0

Pebesma, E. (2018). Simple Features for R: Standardized support for spatial vector data. *The R Journal*, *10*, 439–446. https://doi.org/10.32614/RJ-2018-009

Pilosof, S., Porter, M., Pascual, M., & Kéfi, S. (2017). The multilayer nature of ecological networks. *Nature Ecology & Evolution*, *1*, 0101. https://doi.org/10.1038/s41559-017-0101

R Core Team. (2023). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.r-project.org/

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis* (p. 213). Springer-Verlag. https://ggplot2.tidyverse.org/